

CLAIMS

What is claimed is:

1 1. A method comprising:

2 receiving a group of data having a group of set values;

3 identifying a group of positions of the group of set values within the group of

4 data;

5 for each of the group of positions, encoding a run of non-set values preceding

6 each of the group of positions.

1 2. The method of claim 1 further comprising encoding a second run of non-set

2 values with an ending symbol.

1 3. The method of claim 1 further comprising encoding a group of signs

2 corresponding to the group of set values.

1 4. The method of claim 1 wherein the group of positions are identified without non-

2 loop related conditional branches.

1 5. The method of claim 4 wherein identifying the group of positions without non-

2 loop related conditional branches comprises:

3 performing a logic operation of each value in the group of data; and

4 tracking the group of positions with a result of the logic operation.

1 6. The method of claim 4 wherein the identifying the group of positions without non-

2 loop related conditional branches comprises:

performing a logic operation of each value in the group of data;
setting a flag to a result of the logic operation; and
tracking the group of positions using a conditional instruction with the flag.

7. A method comprising:

loading a group of data into a buffer, the group of data having a group of one or
more set bits;
identifying a position of a first set bit in the buffer without non-loop related
conditional branching;
encoding a run of non-set bits preceding the position; and
shifting the run of non-set bits and the first set bit out of the buffer.

8. The method of claim 7 further comprising encoding a second run of set values
with an ending symbol.

9. The method of claim 7 further comprising encoding a sign bit corresponding to
the first set bit at the position in a second buffer.

10. The method of claim 7 wherein loading a group of data into the buffer comprises:
loading a group of source data into a first buffer, the group of source data
including the group of data;
loading a mask for a bit-plane into a second buffer;
selecting the group of data from the group of source data with the mask; and
extracting the group of data into the buffer.

11. The method of claim 7 wherein identifying the first set bit comprises:

2 shifting each bit of the group of data out of the buffer;
3 comparing each shifted bit with a compare bit to determine if each bit is one of the
4 group of set bits;
5 tracking each of the group of set bits when detected with the comparing; and
6 incrementing a counter for each shifted bit.

1 12. The method of claim 7 wherein the first set bit is identified with one or more look-
2 up tables.

1 13. The method of claim 7 further comprising
2 identifying multiple positions of first set bits in the buffer without non-loop
3 related conditional branching; and
4 simultaneously encoding runs of non-set bits preceding the positions.

1 14. An apparatus comprising:
2 a buffer to host a group of data having a group of one or more set bits; and
3 a variable length coding (VLC) module coupled with the memory, the VLC
4 module to identify the group of set bits' positions and to encode each run
5 of non-set bits with respect to the group of set bits' positions.

1 15. The apparatus of claim 14 further comprising the VLC module to encode a second
2 run of non-set bits with an ending symbol.

1 16. The apparatus of claim 14 further comprising the VLC module to encode a group
2 of sign bits corresponding to the group of set bits.

1 17. The apparatus of claim 14 further comprising a group of one or more look-up
2 tables for the VLC module to identify the group of set bits' positions.

1 18. The apparatus of claim 14 wherein the VLC module to identify each of the group
2 of set bits' positions comprises

3 for each of the group of set bits,

4 to load the group of data into a memory,

5 to flip all bits of the group of data,

6 to set all bits in positions after one of the group of set bits position to non-
7 set bits, and

8 to add the set bits in the memory in an adder tree.

1 19. A machine-readable medium that provides instructions, which when executed by a
2 group of processors of one or more processors, cause said group of processors to perform
3 operations comprising:

4 receiving a group of data having a group of set values;

5 identifying a group of positions of the group of set values within the group of
6 data;

7 for each of the group of positions, encoding a run of non-set values preceding
8 each of the group of positions.

1 20. The machine-readable medium of claim 19 further comprising encoding a second
2 run of non-set values with an ending symbol.

1 21. The machine-readable medium of claim 19 further comprising encoding a group
2 of signs corresponding to the group of set values.

1 22. The machine-readable medium of claim 19 wherein the group of positions are
2 identified without non-loop related conditional branches.

1 23. The machine-readable medium of claim 22 wherein identifying the group of
2 positions without non-loop related conditional branches comprises:
3 performing a logic operation of each value in the group of data; and
4 tracking the group of positions with a result of the logic operation.

1 24. The machine-readable medium of claim 22 wherein the identifying the group of
2 positions without non-loop related conditional branches comprises:
3 performing a logic operation of each value in the group of data;
4 setting a flag to a result of the logic operation; and
5 tracking the group of positions using a conditional instruction with the flag.

1 25. A machine-readable medium that provides instructions, which when executed by a
2 group of processors of one or more processors, cause said group of processors to perform
3 operations comprising:
4 loading a group of data into a buffer, the group of data having a group of one or
5 more set bits;
6 identifying a position of a first set bit in the buffer without non-loop related
7 conditional branching;
8 encoding a run of non-set bits preceding the position; and

9 shifting the run of non-set bits and the first set bit out of the buffer.

1 26. The machine-readable medium of claim 25 further comprising encoding a second
2 run of set values with an ending symbol.

1 27. The machine-readable medium of claim 25 further comprising encoding a sign bit
2 corresponding to the first set bit at the position in a second buffer.

1 28. The machine readable medium of claim 25 wherein loading a group of data into
2 the buffer comprises:

3 loading a group of source data into a first buffer, the group of source data

4 including the group of data;

5 loading a mask for a bit-plane into a second buffer;

6 selecting the group of data from the group of source data with the mask; and

7 extracting the group of data into the buffer.

1 29. The machine-readable medium of claim 25 wherein identifying the first set bit
2 comprises:

3 loading the group of data into a memory;

4 flipping all bits of the group of data;

5 setting all bits after the first zero bit to zero; and

6 adding the set bits in the second memory in an adder tree, the set bits being one
7 bits.

1 30. The machine-readable medium of claim 25 wherein the first set bit is identified
2 with one or more look-up tables.